ECE 538 - Advanced Computer Architecture
Homework Assignment 1 (Individual)

100 points (plus 10 extra credits)

Assigned Date: 09/13/2021 **Due Date:** 09/24/2021

1. **(20 points)** For your research, you profiled an application that requires 900 seconds of execution time on a single-core architecture. You would like to design a multi-core architecture to speed up this particular application. Use Amdahl's Law to answer the following questions:

   (a) (5 points) If you were to rewrite the application to take advantage of a multi-core architecture, what would be the maximum achievable speedup on an architecture with four cores? What would the new execution time be?
   Assuming 800 seconds of the application's execution time is perfectly parallelizable, while 100 seconds must be executed sequentially. Also assume the performance of each core in the quad-core architecture equals that of the original single-core architecture.

   This problem makes use of Amdahl's Law:

   $Et\text{-}new = 800s/4 + 100 = 300s$

   $Speedup = 900s/300s = 3X$

   (b) (5 points) What would be the execution time and speedup achievable with an eight-core machine?

   $Et\text{-}new = 800s/8 + 100 = 200s$

   $Speedup = 900/200 = 4.5X$

   (c) (5 points) What would be the maximum speedup achievable on a machine with unlimited cores?

   Assume the unlimited cores to be ∞

   $Et\text{-}new = 800s/\infty + 100s$, thus $800s/\infty \approx 0$, $Et\text{-}new = 100s$

   $Speedup = 900s/100s = 9X$

   (d) (5 points) What would be the maximum speedup achievable on a machine with unlimited cores if 99% of the execution time could be perfectly parallelized?

   $[\ Et\text{-}new = 900s \times 99\%/\infty + (1-99\%) \times 900s\ ]$

   $Speedup = 900s \Big/ \left[\frac{900s \times 99\%}{\infty} + (1-99\%) \times 900s\right]$

   $= 1/\left[\frac{0.99}{\infty} + (1-0.99)\right] = \frac{1}{0.01} = 100X$

2. **(20 points)** A cryptographic operation takes 1 second to run on a simple embedded processor core. You are considering the design of a coprocessor to accelerate this cryptographic operation in order to improve the energy efficiency of the system. Answer the following questions, using Amdahl's Law where appropriate.

   (a) (5 points) If 95% of the execution time can be accelerated with a coprocessor, what is the maximum speedup theoretically achievable, assuming no computation overlap between the accelerator and the processor?

$$\text{Speedup} = \frac{1}{\frac{0.95}{\infty} + (1-0.95)} = \frac{1}{0.05} = 20\times$$

   (b) (5 points) If you were able to design an accelerator that can execute the aforementioned 95% execution time with a 50x speedup, what would be the maximum achievable speedup of the entire cryptographic operation? What is the new execution time?

$$\text{Speedup} = \frac{1}{\frac{0.95}{50} + (1-0.95)} = 14.493\times$$

$$E_{t\_new} = 1s / 14.493 = 0.069s = 69ms$$

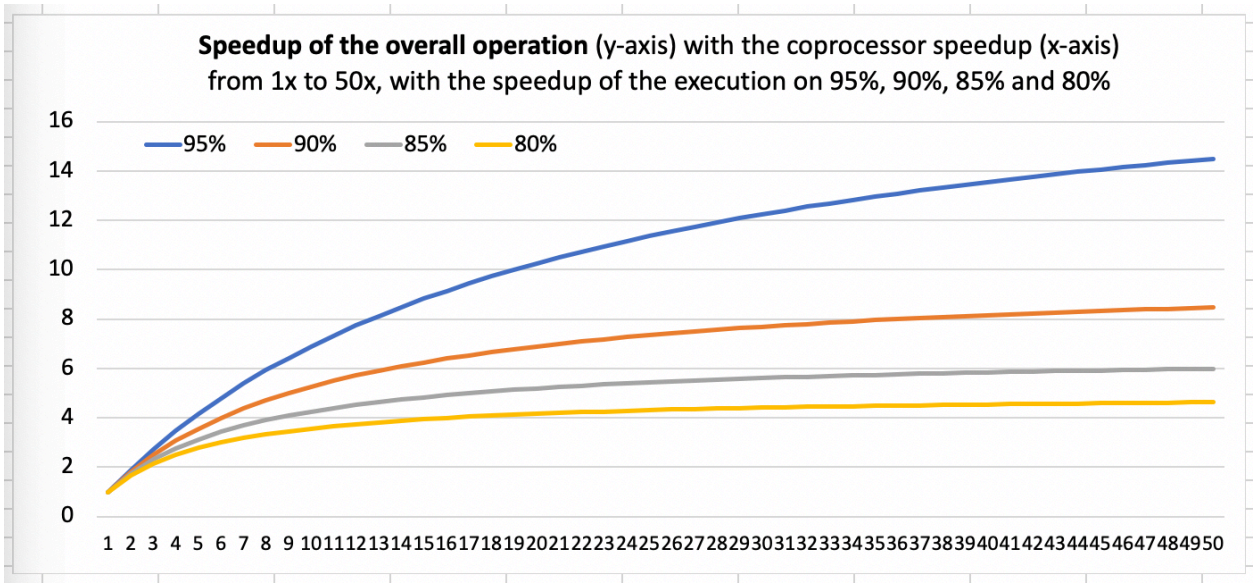
   (c) (5 points) What is the energy improvement of a cryptographic operation with the coprocessor discussed in the previous question if the power of the system doubles with the addition of the accelerator?

Energy = Power × Time

power is doubled w/ the accelerator, but execution time is reduced by 14.493×. thus. Energy Improvement = $\frac{14.493}{2}$

$= 7.25\times$

   (d) (5 points) Plot the speedup of the overall cryptographic operation as the coprocessor speedup varies from 1x to 50x assuming 95% of the execution time can be executed on the accelerator. In the same figure, plot the speedup assuming 80%, 85%, and 90% of the execution time can be executed on the accelerator. Be sure to properly label the axis of your graph and use a legend to differentiate between the four different lines. You may use any graphing tool of your choice.


Speedup of the overall operation (y-axis) with the coprocessor speedup (x-axis)
from 1x to 50x, with the speedup of the execution on 95%, 90%, 85% and 80%
95%
90%
85%
80%
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

3. **(20 points)** Applications are often times categorized as either computation bound or I/O bound. The former spends most of its execution time on computation, while the latter spends most of its time waiting for I/O (i.e. disk, network access, etc.). The following questions address these differences.

   (a) (5 points) Suppose you have an application that spends 70% of its time waiting for disk access and 30% of its time on computation. If you were to run this application on a processor that is 20x faster but with the same I/O speed, what would be the potential speedup?

   this problem makes use of Amdahl's Law

   $$\text{Speedup} = \frac{1}{0.7 + \frac{0.3}{20}} = 1.399\times$$

   (b) (5 points) If instead, you ran the application on a system with the same processor speed but a disk that is 2x faster, what would be the potential speedup?

   $$\text{Speedup} = \frac{1}{\frac{0.7}{2} + 0.3} = 1.54\times$$

   (c) (5 points) Describe how you might go about improving the disk access of this system.

   To improve the disk access, one could use a parallel disk system; Or switch to solid-state disks; Or disk striping ... expand storage.

   (d) (5 points) How else might you improve the speed of the overall system in an I/O bound system? How does that compare to that of a computation bound system?

   Usually, the network time access can be the bottleneck, so speedup the network access might be helpful.

   Compared with that of a computation bound system, the computation bound system can be speeded up by using more cores, new effective cores, more RAM, faster RAM, etc.

4. **(20 points)** Consider a cluster like the one supported by the Center for Advanced Research Computing (CARC) at UNM. Assume the cluster has 500 computers, each of them with a MTTF of 25 days, and the failures follow an exponential distribution and are independent.

   (a) (5 credits) If 1/5 of the computers fail, the cluster is considered to fail. What is the MTTF of the cluster? Under this failure model, does adding more computers increase the MTTF on the cluster? Why?

$$\text{Failure Rate} = 500 \times \frac{1}{25} \times \frac{1}{100} = \frac{1}{5}$$

$$\text{MTTF} = \frac{1}{\text{Failure Rate}} = 5 \text{ (days)}$$

Since we are maintaining a $\frac{1}{5}$ ratio, adding more computers will not change MTTF.

   (b) (10 credits) For the same amount of money, one could buy 800 computers, each with MTTF of 20 days. Assume that the cluster (implementation with either 800 less reliable computers or 500 original computers) is considered to fail if a single computer fails. Repairing the less reliable cluster configuration is 10% less expensive. Which cluster would be better?

For the old system, $\text{MTTF} = 25/500 = 1/20 \text{ (days)} = 1.2 \text{ (hours)}$

For the new system, $\text{MTTF}' = 20/800 = 1/40 \text{ (days)} = 0.6 \text{ (hours)}$

Thereful, the failure rate of the new system is doubled. So, I recommend old system.

   (c) (5 credits) If the cluster is considered to fail if a single computer fails, it can be readily shown that adding more computers does not improve MTTF. List reasons of why we still would like to have clusters with a larger number of computers (at least three reasons, can be related to reliability, performance and power)

① Improve the parallelism on single application.

② Service more users simutaneously

③ For (better) redundancy

④ Fault tolerance

5. **(20 points)** If you are asked to design the power management (PM) policy for processors used to run a realtime application (e.g., streaming a movie on the mobile device). The application should be completed processing one set of data every 30 milliseconds (ms). The processor contains 4 identical cores: if all cores are at the full speed, the processor consumes 2 Watts (W) (0.5 W for each core); if a core is put into sleep model, the leakage power of the core is 10% of the dynamic power; If the entire processor is in sleep, the leakage is 15% of the total dynamic power. Assume that the application is fully parallelizable.

   Consider three PM policies: 1) Use all 4 cores and run at the full speed. Under this policy, application finishes in 20 ms. 2) Use all 4 cores and run the application by setting the voltage and frequency to 3/4 of what are used at the full speed and then sleep if there is slack time till the deadline. 3) Use 3 cores while letting the 4^{th} core sleep.

   (a) (10 credits) Can policy 2 and 3 meet the deadline requirement of the application? Give the reason.

   Use policy 2: $Et = 20 \times 4/3 = 26.67\ ms < 30\ ms$ ✓

   Use policy 3: $Et' = 20 \times \frac{1}{3/4} = 20 \times 4/3 = 26.67\ ms < 30ms$ ✓

   Both of the times are under 30 ms, so policy 2 and 3 can meet deadline.

   (b) (10 credits) Which option is more energy efficient and why? Which one requires higher peak power (i.e., the maximum power)

   Use policy 1: $P_1 = 2w.$ $E_1 = 2w \times 20ms + (30-20)ms \times 2w \times 15\% = 43$

   Use policy 2: $P_2 = 2w \times (\frac{3}{4})^2 \times \frac{3}{4} = 2w \times (\frac{3}{4})^3 = 0.844\ w.$ [since $P \propto V^2 f.C.$]

   $E_2 = 2w \times (\frac{3}{4})^3 \times 20 \times \frac{4}{3} + (30 - 20 \times \frac{4}{3}) \times 2w \times 0.15 = 23.5$

   (c) (10 extra credits) Find a policy that leads to the minimum total energy, assuming that no dynamic voltage scaling (i.e., changing the voltage supply level during run time) is allowed. Give the details.

use Policy 3:

$P_3 = 3 \times 0.5 + 0.5 \times 10\% = 1.55w$

$E_3 = 1.55 \times 20 \times \frac{1}{3/4} + (30 - 20 \times 1/3/4) \times 2w \times 15\%$

$= 42.33.$